

## Research Article

# A New Real-Time Path Planning Method Based on the Belief Space

**Yu-xin Zhao, Xin-an Wu, and Yan Ma**

*College of Automation, Harbin Engineering University, Harbin 150001, China*

Correspondence should be addressed to Yu-xin Zhao; zhaoyuxin@hrbeu.edu.cn

Received 5 October 2013; Accepted 28 October 2013

Academic Editor: Xiaojie Su

Copyright © 2013 Yu-xin Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new approach of real-time path planning based on belief space is proposed, which solves the problems of modeling the real-time detecting environment and optimizing in local path planning with the fusing factors. Initially, a double-safe-edges free space is defined for describing the sensor detecting characters, so as to transform the complex environment into some free areas, which can help the robots to reach any positions effectively and safely. Then, based on the uncertainty functions and the transferable belief model (TBM), the basic belief assignment (BBA) spaces of each factor are presented and fused in the path optimizing process. So an innovative approach for getting the optimized path has been realized with the fusing the BBA and the decision making by the probability distributing. Simulation results indicate that the new method is beneficial in terms of real-time local path planning.

## 1. Introduction

Recently, the development and application of autonomous robots are with growing interest in industrial and military fields. As we all know, navigation is one of the key technical problems for autonomous robots, and the most important factor of navigation is map building based on the sensor system, especially when the autonomous robots are working in an entire unknown environment. The environment is reconstructed by merging the information transferred from the sensor system during the motion. To build a practical map, one of the most difficult problems is due to the poor environment information of the sensor system which has inherent wide radiation cone and the phenomenon of multiple reflections. Thus, how to describe these uncertainties and filter out inaccurate and conflicting information and how to construct the environment view are the hot issues.

In these few years, there are about three types of approaches of constructing the environment view that appeared in exoteric literatures. The first type is the occupancy grid mapping method [1], which represents maps with fine-grained grids that model the occupied and free space of the environment. The second type is the geometrical information mapping method [2], which uses some sets of

line, angles and polygons to describe the geometry of the environment. The third way is the topological method [3, 4], which models the environment by a series of landmarks that are connected via arcs.

In order to describe the uncertainties, or filter out the conflicting information detected by sensors, the probabilistic algorithms [5] was proposed by a definitive formulation through the Bayesian technique originally. Then a family of algorithms [6] based on fuzzy theory [7] established the uncertainty information model in each cell. In a similar way, another way based on Dempster-Shafer theory described the uncertainty model by using the belief functions. In these years, the neural network technique have been introduced with using the learning ability of the neural cell [8].

There is no doubt that the optimization problem is quite important for autonomous robots path planning. So many evolutionary optimizing techniques like genetic algorithm [9–11], neural network [12], and ant colony optimization [13] are extensively used in solving the global path planning problems, on condition that the environment has been detected. But these algorithms do not work in a real-time local path planning environment, because, besides the path length, some other factors such as the underwater robot's self-characters and the influence of the special environment

(ocean current, wind speed) also influence the selection of the local target point in real-time local detecting path planning. As far as we know, few researchers consider these factors in solving real-time path planning problems.

In this paper, a novel real-time path planning approach based on the belief space is introduced. As the transferable belief model (TBM), which is popular in these years, can be used to describe a highly flexible model to manage the uncertainty information in the multisensor data fusion problems. In particular, many applications of TBM have been presented in mobile vehicles and other areas [14–17].

The rest of the paper is organized as follows. In Section 2, the uncertainty model of the sensor detection is shortly described, and the main idea of the transferable belief model is written in Section 3. In Section 4, the complex environment information is expressed by the double-safe-edges free space, which can simplify the real-time detecting environment information and prepare for the real-time path planning. In Section 5, the belief space is established according to the belief functions of the factors that affect the selection of the local target points, so the optimization local target point can be found at each step. The connection line of these optimization local target points is the optimization path of the task. Section 6 shows the experimental results of the new path planning approach and Section 7 comprises of conclusion.

## 2. Uncertainty Model of the Sensor Detection

Sonar is far from being an ideal sensor, mainly due to the width of the radiation cone and to the multiple reflections phenomenon. The former does not allow determining the exact angular position of the obstacle on the fixed angle  $\theta$  arc of the circle corresponding to the detected distance. The latter needs a more thorough explanation. The sonar waves are reflected in two different ways depending on the surface irregularities. If their sizes are much smaller than the wavelength of the signal, we have a diffused reflection; that is, the incident energy is scattered in all directions; otherwise, the reflection is mainly specular and the beam may either reach the receiver after multiple reflections or even get lost [18].

The uncertainty model has been set up by fuzzy measure approach. A single reading  $r$  provides the information that one or more obstacles are located somewhere along the  $\theta$  arc of circumference of radius  $r$ . Hence, there is evidence that points located in the proximity of this arc “occupied.” On the other hand, points well inside the circular sector of radius  $r$  are likely to be “empty.” To model this knowledge, we introduce the two functions [19]:

$$f_e(\rho, r) = \begin{cases} k_e & 0 \leq \rho \leq r - \Delta r \\ k_e \left( \frac{r - \rho}{\Delta r} \right)^2 & r - \Delta r \leq \rho \leq r \\ 0 & \rho \geq r, \end{cases} \quad (1)$$

$$f_o(\rho, r) = \begin{cases} 0 & 0 \leq \rho \leq r - \Delta r \\ k_o \left[ 1 - \left( \frac{r - \rho}{\Delta r} \right)^2 \right] & r - \Delta r \leq \rho \leq r \\ 0 & \rho \geq r + \Delta r. \end{cases} \quad (2)$$

That describe, respectively, how the degree of certainty of the assertions “empty” and “occupied” varies with  $\rho$  for a given range reading  $r$ . Here,  $\rho$  is the distance from the sensor  $k_e$  and  $k_o$  are two constants corresponding to the maximum values attained by the functions, and  $2 \times \Delta r$  is the width of the area considered “proximal” to the arc of radius  $r$  [20].

Since the intensity of the waves decreases to zero at the borders of the radiation cone, the degree of certainty of each assertion is assumed to be higher for points close to the beam axis. This is realized by defining an angular modulation function [19]:

$$f_a(\vartheta) = \begin{cases} D(\vartheta) & 0 \leq |\vartheta| \leq \gamma \\ 0 & |\vartheta| > \gamma, \end{cases} \quad (3)$$

$$g_d(\rho) = 1 - \frac{1 + \tanh(2(\rho - \rho_v))}{2}. \quad (4)$$

In order to weaken the confidence of each assertion as the distance from the sensor increases, the parameter  $\rho_v$  plays the role of a “visibility radiuses,” where a smooth transition occurs from certainty to uncertainty. The motivation for introducing this function is twofold. Firstly, as the possibility of multiple reflections increases as the beam makes a loner fly. Besides, narrow passages appear to be obstructed if seen from a large distance, due to the sensor wide radiation angle. By varying the visibility radius according to the characteristics of the environment, it is possible to obtain a more correct detection behavior [20].

## 3. The Transferable Belief Model (TBM)

TBM is a model for describing quantified beliefs based on belief function. Beliefs can be held at two levels: (1) a “credal” level where beliefs are entertained and quantified by belief functions; (2) a “pignistic” level where beliefs can be used to make decisions and are quantified by probability functions. The relation between the belief function and the probability function when decisions must be made is derived and justified [21].

In TBM, the actual value  $\omega_0$  of the variable whose finite domain is a given set  $\Omega$  has been considered. A basic belief mass (BBM) denoted by  $m^\Omega$  is used to represent the uncertainty about the value of  $\omega_0$ .  $m^\Omega(A)$ , which is the basic belief assignment (BBA), is given to  $A \subseteq \Omega$ .  $m^\Omega$  maps  $2^\Omega$ , the power set of  $\Omega$  on  $[0, 1]$ , and satisfies [22]:

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1. \quad (5)$$

The mass  $m^\Omega(A)$  represents the part of belief that supports that the actual value  $\omega_0$  belongs to  $A$  and without more specific several useful functions [23].

Belief function is defined as

$$\text{bel}^\Omega(A) = \sum_{B: \emptyset \neq B \subseteq A} m^\Omega(B). \quad (6)$$

The value  $\text{bel}^\Omega(A)$  represents the total amount of belief supporting that  $\omega_0$  is in  $A$  or without supporting that it is in  $\bar{A}$  where  $\bar{A}$  is the complement of  $A$  relative to  $\Omega$ .

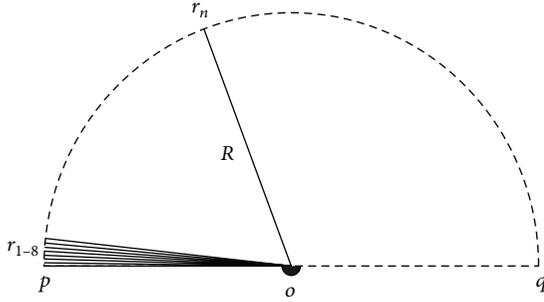


FIGURE 1: The sketch map of the sensor detection simulation.

Plausibility function is defined as

$$pl^\Omega(A) = \sum_{B:\theta \cap B \neq \emptyset} m^\Omega(B). \quad (7)$$

The value  $pl^\Omega(A)$  represents the total amount of belief supporting that  $\omega_0$  might be in  $A$  without supporting that it might be in  $\bar{A}$ .

Combination rules: in general Bayesian theorem, the sensor detection  $x$  is the vector of plausibility  $pl^X[h_i](x)$  for all  $h_i \in H$ . The conditional belief can be written by probability function  $pl^X[h_i](x) = P^X[h_i](x)$ . It is easier to use the likelihood of  $h_i$  given  $x$ , denoted by  $l(h_i | x)$  [24]. So given the likelihood  $l(h_i | x)$  for every and for every  $h_i \in H$ , Smets [25] has proved

$$m^H[x](A) = \prod_{h_i \in A} l(h_i | x) \prod_{h_i \in \bar{A}} (1 - l(h_i | x)), \quad (8)$$

$$pl^H[x](A) = 1 - \prod_{h_i \in A} (1 - l(h_i | x)).$$

Decision making function is defined as

$$Bet P^\Omega(A) = \sum_{B \subseteq \Omega} \frac{|A \cap B|}{|B|} \frac{m^{\Omega(B)}}{1 - m^{\Omega(\emptyset)}} \quad \forall A \subseteq \Omega. \quad (9)$$

In the TBM, when a decision has to be made, a probability functions  $Bet P^\Omega(A)$  on  $\Omega$  must be adopted.  $Bet P^\Omega$  is a probability measure.

#### 4. The Simulation of the Process of the Sensor Detection

4.1. *The Process of Detecting of the Sensor.* We will build a simulation environment about the detection process of the robot sensor for testing the new approach of the real-time path planning process. The robot sensor is an initiative sensor, the angle of the detecting is  $180^\circ$ , and the distance of detecting is  $R$ , so this paper will make 180 lines which starts from the particle of the robot and the length are  $R$  and the angle of each line is  $1^\circ$ .

In Figure 1, point  $o$  is the particle of the sensor, the sector  $\widehat{poq}$  is the detecting area of the sensor, the distance

$R$  is the max distance of detecting, the lines  $or_n = (n \in [0, 180])$  are the sound wave of the sensor, and the diameter of the sector and the  $y$ -axis of the robot is vertical. Thus, with this enactment, after each detecting of the environment, the environment information is the 181-distance, information in the sector  $\widehat{poq}$ ; they are the position information of the obstacles.

Figure 2 shows the four-detail process of the simulation of the detecting process of the sensor; in Figure 2(a) is the state of the no obstacle at time  $t_0$ ; it gives the particle of the sensor, the 181 lines of  $R$  distance, and the sector area of the detecting; in Figure 2(b) is obstacles which the sensor needs to detect at time  $t_s$ ; in Figure 2(c) is the detection state of the sensor has detected the obstacles in Figure 2(b); it shows that some of these 181 lines have been cut in this state, so the process of detecting has been built; Figure 2(d) shows the result of the detection; the position information of the obstacles can be noted by the set  $Q_t (t = 0, 1, \dots, n)$ .

4.2. *The Transformation of the Detecting Space Coordinates of the Sensor.* In the process of the path planning, the position information of the obstacle, and the robot, the information of the whole target point and the local target points must be described at each time, so it needs a uniform reference frame. There are two reference frames in the process of path planning of this paper: the reference frame of the robot movement and the reference frame of the sensor detection, so it needs the transition of the reference frame. In this paper, the reference frame of the robot movement is a vertical coordinate; the start point  $S$ , the whole target point  $G$ , and the position of robot  $O$  can be denoted; the reference frame of the sensor detection is a pole that coordinates the obstacle information  $Q$  and the local target point can be denoted, and the transition of these two coordinates is in Figure 3.

In Figure 3, the origin of the vertical coordinate is  $O_g$ , the position of the robot and the whole target point is  $o$  and  $G$ , the origin of pole coordinate is  $o$ , vector  $op$  is the pole axes, and the angle between the pole axes and the  $x$ -axis of the vertical coordinate is  $\beta$  so the coordinate of the obstacles or the local target point is  $(\rho_n, \theta)$ . The vertical coordinate in the reference frame of the sensor detection is  $(x_p, y_p)$ :

$$x_p = \rho \cos \theta, \quad (10)$$

$$y_p = \rho \sin \theta.$$

So the vertical coordinate in the reference frame of the robot movement is the position vector which is  $[x_0, y_0, z_0]^T$ :

$$x = x_p \cos \beta_0 - y_p \sin \beta_0 + x_0, \quad (11)$$

$$y = x_p \sin \beta_0 + y_p \cos \beta_0 + y_0.$$

#### 5. The Procedure of Confirming the Double Safe Edges Free Space

5.1. *The Description of the Environment Information in Real-Time.* In an uncertainty and dynamic environment, the environment information for path planning is obtained from

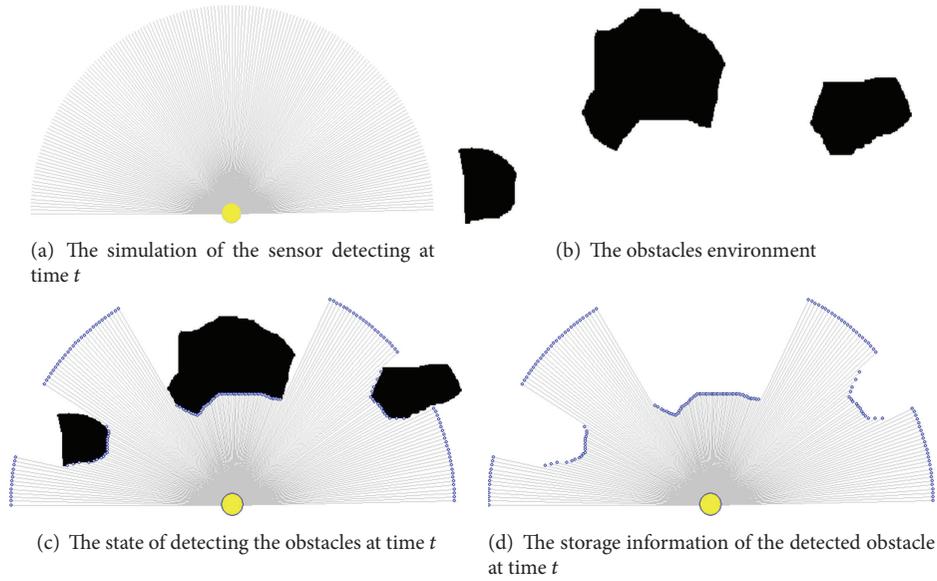


FIGURE 2: Simulation results of the sensor detection at time  $t$ .

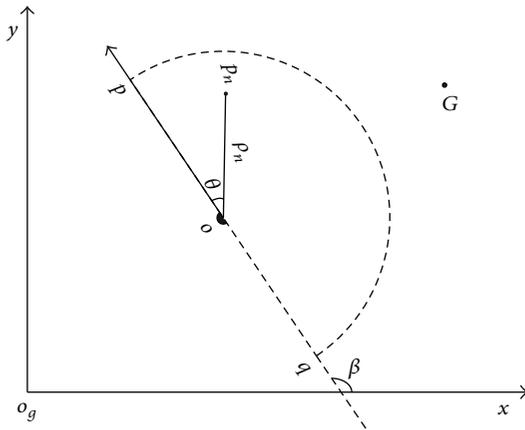


FIGURE 3: The transformation of the detecting space coordinates of the sensor.

the sensor on the robot only, so the algorithm should have well real-time ability and it is also the first step of generating the robot's motion. According to the sensor detecting model, we propose a method for searching the important information from the detecting information in this paper, which is called the double-safe-edges (DSE) information.

**5.1.1. Searching for the Sensor Edges.** Obviously, the sensor edges can be searched directly from the sensor detecting information, and its distance and direction can be ensured according to the position of the obstacles.

In Figure 4, point  $o$  is the particle of the robot (the sensor and robot at the same particle) and the self-safe area of the robot is a circle whose radius is the particle of the robot, and the radius is  $r$ , the range of the angle is  $\pi/2$ , the biggest detection radius is  $R_{\max}$ , the obstacles are  $ob_1$  and  $ob_2$ , and the safe distance between the robot and obstacle is  $d_i$ . Because

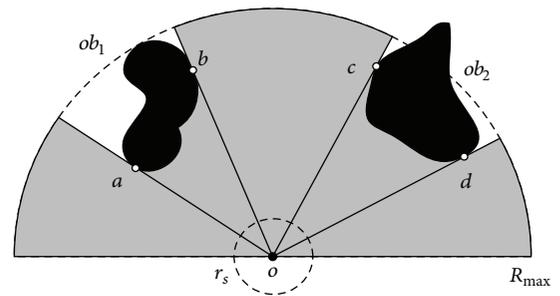


FIGURE 4: The sketch map of searching the sensor edges at time  $t$ .

the robot detection area is a hemicycle in front of the robot, so we use the lines to simulate the detecting process and the lines' length is  $R_{\max}$ , the number is 180, and the angle of them is  $1^\circ$ . So it can find the sensor edges set  $\{a, b, c, d\}$  quickly according to the decision parameter  $E_i$ ,  $E_i = R_{\max} - d_i$ ,  $i \in [0, 180]$ . The rule of detecting sensor edges is as follows: if  $E_i \geq 2r_s$ , so the sensor edges are appearance. The rule of detecting the direction of sensor-edges is as follows: suppose the searching direction of the sensor-edges from the left of the robot, if  $\min\{E_{i-1}, E_{i+1}\} = E_{i-1}$ , so the direction of  $P_i$  is left, denoted by  $L$ , if  $\min\{E_{i-1}, E_{i+1}\} = E_{i+1}$ , so the direction of  $P_i$  is right, denoted by  $R$ .

In Figure 5, it is the state curve of  $E$  at certain time, the sensor edges set is  $\{a, b, c, d\}$ , and the direction set is  $\{R, L, R, L\}$ .

**5.1.2. Searching for the Double-Safe-Edges.** The edges are based on the sensor as mentioned above. But the robot has its own safe area because of its special shape and kinematics, if it considers the sensor edges only, and the path planning must be failing. So it is necessary to consider the environment information and the robot's safe area together.

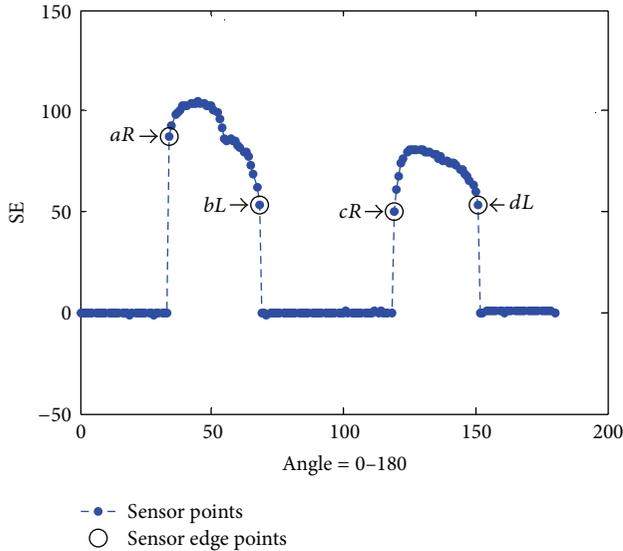


FIGURE 5: The result of the simulation of searching the sensor edges at time  $t$ .

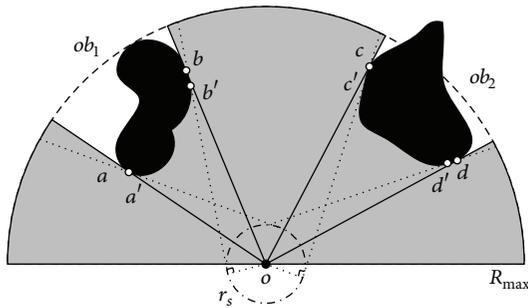


FIGURE 6: The sketch map of searching the double-safe-edges points at time  $t$ .

In this paper the definition of the double-safe-edges has consider the environment information and robot's safe area.

**Definition 1** (double-safe-edges (DSE)). When the sensor edges been found, the algorithm will search some points which considering the environment information and robot's safe area, searching start from the sensor-edges according to its directions, the tangent lines which from these points to the robot's safe circle are tangent to the edges of the obstacles at the same time. The robot and obstacle are at the different sides of the tangent line. These points are the set of the double-safe-edges points and these lines are the set of the double-safe-edges.

In Figure 6, point  $o$  is the particle of the robot, the set of sensor-edges  $\{a, b, c, d\}$ , and the set direction  $\{R, L, R, L\}$ ; the radius of the safe circle of the robot is  $r_s$ . Figure 7 shows that the state curve of  $E$  at certain time, the set of double-safe-edges  $\{a', b', c', d'\}$ , and the set of direction  $\{R, L, R, L\}$  can be found according to the definition of the double-safe-edges. This double-safe-edges information is very important to generate the motion commands in this paper.

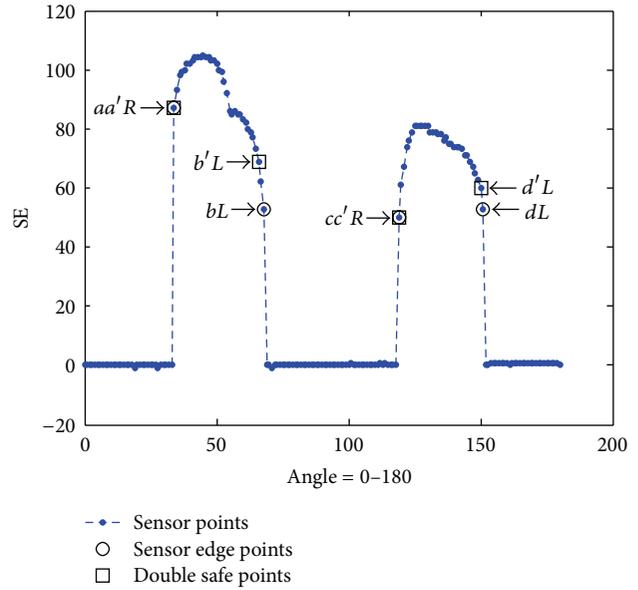


FIGURE 7: The result of the simulation of searching the double-safe-points at time  $t$ .

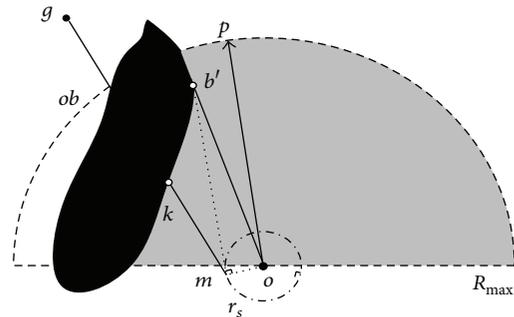


FIGURE 8: The sketch map of S-DSE at time  $t$ : DSE points set is  $\{b'\}$ , safe distance set is  $\{mb'\}$ , and safe distance set is  $op$ .

The success of finding the double-safe-edges means that the environment detected by the sensor in real-time has been analyzed and interpreted efficiently, the environment information has been simplified, the real-time has been increased, and the robot's safe area and the kinematics have been considered, so it will be efficient in generating the motion commands at the next step.

**5.2. The Types of the Double-Safe-Edges.** There are three types of the double-safe-edges: S-DSE, M-DSE, and Z-DSE.

(1) S-DSE. There is only single DSE point after analyzing and interpreting the environment information. In Figure 8, line  $mg$  connects the goal  $g$  and point  $m$ ; if  $k$  is the interaction point of  $mg$  and obstacle, the robot must escape the obstacle. So the robot's safe moving direction is  $\angle_s = \angle_{op}$ , and the safe

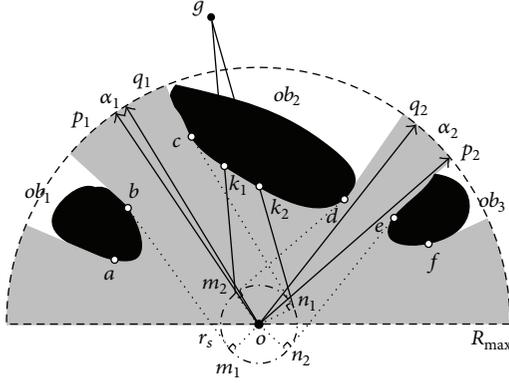


FIGURE 9: The sketch map of S-DSE at time  $t$ : DSE points set is  $\{a, b, c, d, e, f\}$ , safe distance set is  $\{m_1b, n_1c, m_2d, n_2e\}$ , and safe distance set is  $\{op_1, oq_1, op_2, oq_2\}$ .

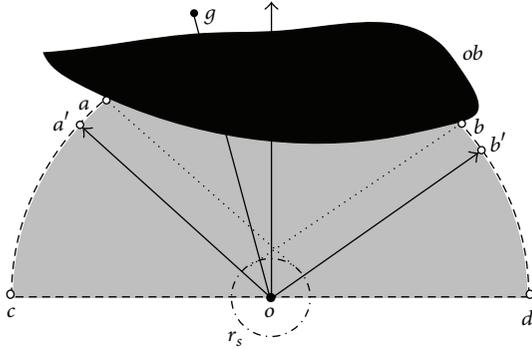


FIGURE 10: The sketch map of S-DSE at time  $t$ : DSE points set is  $R_{\max}$ , and safe distance set is  $\{\emptyset\}$ , and safe distance set is  $\{oa', ob'\}$ .

moving distance is  $d_{e-e} = mb'$  according to the DSE point  $b'$  and line  $mb'$ ,  $op//mb'$ .

(2) *M-DSE*. There is more than one DSE point after analyzing and interpreting the environment information. In Figure 9, lines  $m_2g$  and  $n_1g$  connect target point  $g$  and the interaction points  $m_2g$  and  $n_1g$  which are on the safe circle; if  $k_1$  and  $k_2$  are the interaction points of  $m_2g$ ,  $n_1g$ , and obstacle, the robot must escape the obstacle. So the possible path set is  $\{op_1, oq_1, op_2, oq_2\}$  according to the DSE points set  $\{a, b, c, d\}$  and DSE lines set  $\{m_1b, n_1c, m_2d, n_2e\}$  and  $\alpha_1$  and  $\alpha_2$  are the angle vector  $op_1$  and  $oq_1, op_2$  and  $oq_2$ .

(3) *Z-DSE*. There is zero DSE point after analyzing and interpreting the environment information. The first situation is that there is no obstacle around the robot, so the robot can move to the target directly; the second situation is that the part or whole of detection area that has been enveloped by the obstacle. For example, In Figure 10, there is part detection area has been enveloped by the obstacle. In this situation, there is zero DSE, so the robot enters the state of cruising in order to find the DSE. The rule of cruising is that if part of detection area  $cd$  has been enveloped, so the directions set  $\{oa', ob'\}$  will be found according to the points set  $\{a, b\}$  and the moving direction  $oc$  will be found according to

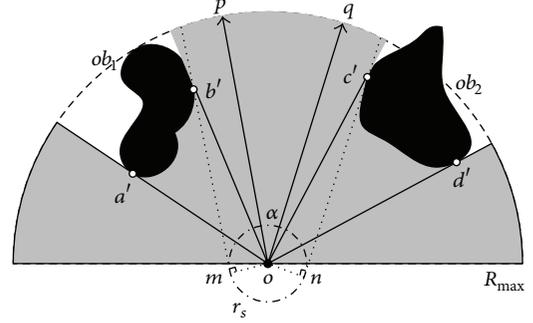


FIGURE 11: The sketch map of the safe distance and direction of the double safe edge points at time  $t$ .

the heuristic algorithm. So the robot can move according to this rule and find the DSE at the same time until the DSE appear.

5.3. *The Description of the Double-Safe-Edges Free Space*. After searching the double-safe-edges, the algorithm has transformed the focus from the environment information to some double-safe-edges points, and these points can be used to generate the robot's motion. We will describe the double-safe-edges free space in this part.

In Figure 11,  $b'$  and  $c'$  are the double-safe-edges points. Lines  $ob'$  and  $oc'$  are the distances from the particle to the double-safe-edges point, and points  $m$  and  $n$  are on the circle of the safe area of the robot,  $om = on = r_s$ . Lines  $mb'$  and  $nc'$  are the distances of the tangent lines,  $mb' = \sqrt{ob'^2 - om^2}$ ,  $nc' = \sqrt{oc'^2 - on^2}$ , and  $op//mb'$ ,  $oq//nc'$ ,  $op$ , and  $oq$  are the possible planning distances of navigation. So the double-safe-edges free space can be defined by the sector area  $poq$ . This area is a free moving space and the robot can select the local target point according to some rule to finish the motion command on time.

## 6. The Optimization of the Real-Time Local Path Planning Based on the Belief Space

6.1. *The Description of the Path Optimization in Real-Time Path Planning*. Although the environment information can be detected by the sensor in real-time, the robot did not know the whole environment information; thus, optimizing the whole path in real-time path planning cannot come true. But there are still some important factors to affect the selection of the path in local environment, and we consider the six local planning factors in this paper, the avoidance collision factor  $R$  between the robot and the obstacles, the kinematics factor  $M$ , the self-safe area factor  $S$ , the path length factor  $L$ , moving obstacle factor  $B$ , and other factors (ocean current, wind speed, and so on)  $C$ , and this part will analyse the influence of the  $R$ ,  $B$ , and  $C$  in real-time local detection planning.

In Figure 12, the robot's position is  $o_t$  at time  $t$ , the real-time detection space is  $W_t$ , the local target set is  $G_t$ ,  $G_t \subseteq W_t$ , and the target is  $g$ . It is the sketch mapping of analyzing the optimization in local detection space  $W_t$ .

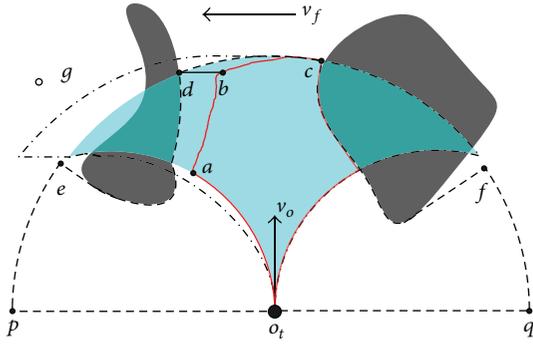


FIGURE 12: The sketch map of the analysis of the optimization in local path planning at time  $t$ .

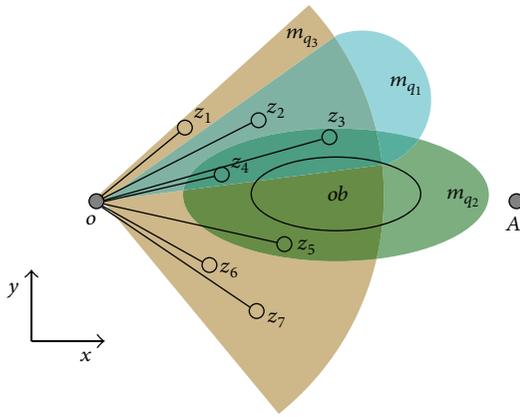


FIGURE 13: The sketch map of the base idea of the belief space at time  $t$ .

Firstly, it can find the edge area (the broken line area)  $B_t$ ,  $B_t \subseteq W_t$ , according to the factor  $R$ , and  $B_t$  is the selected area of  $G_t$ ; secondly, supposing that the robot's kinematics is the gyration movement, so the reachable area (the undertone area)  $M_t$ ,  $M_t \subseteq W_t$  according to the factor  $M$ , and finally, these two factors can make the local goal selection area smaller. Supporting the other factor  $C$  can make the robot have the speed  $v_f$ , and this speed can make the displacement  $s$ ,  $s = v_f \cdot t$ . So it can find the reachable area (the real line area)  $F_t$ ,  $F_t \subseteq (B_t \cap M_t) \subseteq W_t$ , at time according to the factors. This area will be smaller when the consideration factors increase, so the analysis treating and fusing these factors is a necessary method to optimize the path in real-time local detection path planning.

**6.2. The Original Idea of the Belief Space in Local Target Selection.** In Figure 13, it is the selection local target point situation in which the robot  $O$  must reach the target point  $A$  according to some selection rules and the local target point set is  $Z = \{z_1, z_2, \dots, z_7\}$  and the selection rules set is  $Q = \{q_1, q_2, \dots, q_j\}$ , so it needs to fuse these selection rules in order to find the optimization local target point.

We note that the selection state space is  $H = \{\text{null}, \text{select}, \text{delect}, \text{unknown}\}$ , described by  $H = \{\emptyset, S, R, U\}$ ,  $U \neq \emptyset$ . For each  $z_i \in Z$ , the selection state space  $c_i \in H$  is known and

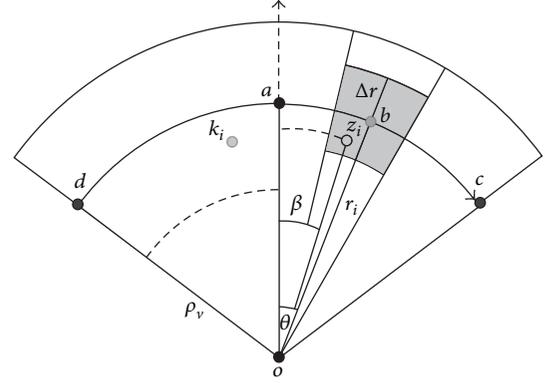


FIGURE 14: The sketch map of the belief function distribution of the sensor detection at time  $t$ .

the influence from  $q_j$  to  $z_i$  can be defined by the BBA in the selection state space  $H$ , denoted by  $m_{q_1}^H(z_i)$ , so the BBA set in  $H$  is  $M = \{m_{q_1}^H(z_1), m_{q_1}^H(z_2), \dots, m_{q_1}^H(z_7)\}$ . Three gray areas are the area of the BBA set  $\{m_{q_1}^H, m_{q_2}^H, m_{q_3}^H\}$ , denoted by  $\{X_{m_{q_1}}^H, X_{m_{q_2}}^H, X_{m_{q_3}}^H\}$ , and these are also the descriptions of the obstacle information, kinematics, and the path length factors, so the belief space can be defined, denoted by  $X_m^H = X_{m_{q_1}}^H \cup X_{m_{q_2}}^H \cup X_{m_{q_3}}^H$ , the local target points' belief can be defined by the belief functions, and the belief functions can be fused according to the TBM rules. The definition of the fusing is

$$\begin{aligned} m^H(z_1) &= m_{q_3}^H(z_1), \\ m^H(z_2) &= m_{q_1}^H(z_2) \odot m_{q_3}^H(z_2), \\ m^H(z_3) &= m_{q_1}^H(z_3) \odot m_{q_2}^H(z_3) \odot m_{q_3}^H(z_3), \\ m^H(z_4) &= m_{q_1}^H(z_4) \odot m_{q_2}^H(z_4) \odot m_{q_3}^H(z_4), \\ m^H(z_5) &= m_{q_2}^H(z_5) \odot m_{q_3}^H(z_5), \\ m^H(z_6) &= m_{q_3}^H(z_6), \\ m^H(z_7) &= m_{q_3}^H(z_7). \end{aligned} \quad (12)$$

The selection of the local target point must satisfy every belief function distribution at the same time, so some local target points can be deleted and the set has been changed to  $\{m^H(z_3), m^H(z_4)\}$ , and the optimization local target point  $Z_3$  can be found if the fusing belief distribution is  $m^H(z_3) > m^H(z_4)$ ; thereby the selection of local target point at certain time in belief space has been finished and the aim of optimization came true.

### 6.3. The Method of Making the Belief Function

**6.3.1. The Belief Function Distribution of the Sensor Detection.** As the uncertainty model has been described, we further discuss the belief function distribution in Figure 14. The angle

of detection is  $90^\circ$ , the direction of  $oa$  is the coordinate axes, and the right is positive. The coordinate of position is  $(r_i, v)$ , and from (1)–(4), the detection distance is  $r_i$ , the angle between  $b$  and  $oa$  is  $\theta$ , the uncertainty area because of the detecting of  $b$  is defined by the  $\gamma_b = 12.5^\circ$  sector and the  $2 \cdot \Delta r = R/4$  width approach to the detection distance, and this area is the gray area in Figure 14. The point  $z_i$  is in this area and the angle between  $z_i$  and  $oa$  is  $\beta$ , so the angle between  $z_i$  and  $b$  is  $\alpha = |\theta - \beta|$ , and the detection area uncertainty function is

$$f_a(\theta) = \begin{cases} D(\theta) & 0 \leq |\theta| \leq 45^\circ \\ 0 & |\theta| > 45^\circ, \end{cases}$$

$$g_a(\rho) = 1 - \frac{1 + \tanh(2|\rho - \rho_v|)}{2}, \quad (13)$$

$$\left(0 \leq \rho \leq R, \rho_v = \frac{R}{2}\right).$$

The detection position uncertainty function is

$$f_b(\alpha) = \begin{cases} D(\alpha) & 0 \leq |\alpha| \leq 45^\circ \\ 0 & |\alpha| > 45^\circ, \end{cases}$$

$$g_b(\rho) = 1 - \frac{1 + \tanh(2|\rho - \rho_v|)}{2}, \quad (14)$$

$$\left(-\Delta r \leq \rho \leq \Delta r, \rho_v = \pm \frac{\Delta r}{2}\right).$$

According to the analysis mentioned above, the detection area uncertainty distribution is from the coordinate axes to the opposition side, and the detection position uncertainty distribution is from  $\pm 12.5^\circ$  sector to the  $\pm \Delta r$  distance area. So the point  $k_i$  in the detection area plausibility function is defined as

$$pl_o^H[x_k](T) = f_{ak}(\theta) g_{ak}(\rho). \quad (15)$$

The point  $z_i$  in detection position plausibility function is defined as

$$pl_{ob}^H[x_z](T) = f_{as}(\theta) g_{as}(\rho) f_{bz}(\alpha) g_{bz}(\alpha). \quad (16)$$

So the position  $z_i$  “occupy” and “empty” plausibility function can be defined as

$$pl_{ob}^H[x_z](T, o) = f_{az}(\theta) f_{az}(\alpha) g_{bz}(\alpha) f_o(\rho, r_i),$$

$$pl_{ob}^H[x_z](T, e) = f_{az}(\theta) f_{az}(\alpha) g_{bz}(\alpha) f_e(\rho, r_i), \quad (17)$$

$$pl_{ob}^H[x_z](T, (o, e)) = 1 - \prod_{h_i \in (o, e)} (1 - l(h_i | x_z)).$$

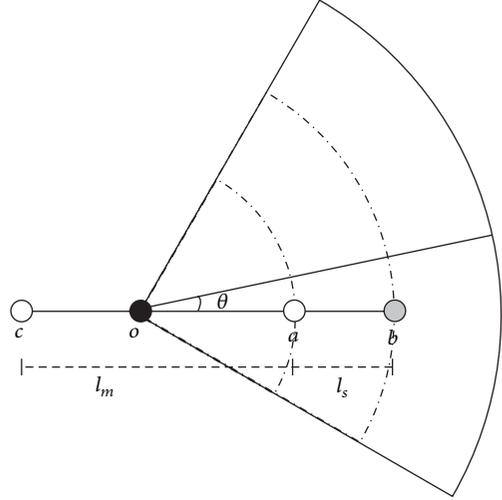


FIGURE 15: The sketch map of the belief function distribution of the safe distance at time  $t$ .

Thus, the BBA of the “occupy” and “empty” in TBM can be defined as

$$m_{ob}^H(o) = \prod_{h_i \in o} l(h_i | x_z) \prod_{h_i \in \bar{o}} l(h_i | x_z),$$

$$m_{ob}^H(e) = \prod_{h_i \in e} l(h_i | x_z) \prod_{h_i \in \bar{e}} l(h_i | x_z), \quad (18)$$

$$m_{ob}^H(o, e) = \prod_{h_i \in (o, e)} l(h_i | x_z) \prod_{h_i \in (0, e)} l(h_i | x_z).$$

This BBA space is defined in the belief space. They are the belief functions distribution according to the base idea of the belief space, denoted by  $\{X_{ob}(i)\}$ .

**6.3.2. The Belief Function Distribution of the Safe Distance to the Obstacle.** Figure 15 shows the belief function distribution of the safe distance to the obstacle, and point  $o$  is the particle of the robot, point  $b$  is on the edge of the obstacle, the shortest safe distance is  $l_s$ , and it is the radius of the self-safe area. In a real environment, when the robot enters into a specified distance (alertness distance)  $l_m$ , it needs to calculate the dangerous degree of collision. So the safe distance function can be given as

$$f_t(\rho) = \frac{\rho - l_s}{l_m} \quad (l_s < \rho < l_m). \quad (19)$$

Then the “safe” and “dangerous” plausibility function can be defined as

$$pl_{am}^H[x_t](T, s) = pl_o^H[x_b](T) f_t(\rho),$$

$$pl_{am}^H[x_t](T, d) = pl_o^H[x_b](T) (1 - f_t(\rho)), \quad (20)$$

$$pl_{am}^H[x_t](T, (s, d)) = 1 - \prod_{h_i \in (s, d)} (1 - l(h_i | x_t)).$$

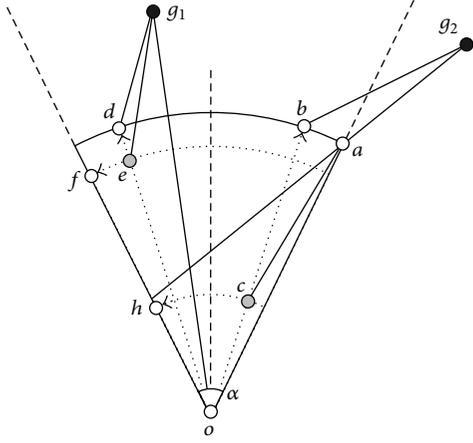


FIGURE 16: The sketch map of the belief function distribution of the optimization the path at time  $t$ .

Finally, the BBA of the “safe” and “dangerous” can be defined as

$$\begin{aligned} m_{am}^H[x_t](s) &= \prod_{h_i \in s} l(h_i | x_t) \prod_{h_i \in \bar{s}} (1 - l(h_i | x_t)), \\ m_{am}^H[x_t](d) &= \prod_{h_i \in d} l(h_i | x_t) \prod_{h_i \in \bar{d}} (1 - l(h_i | x_t)), \\ m_{am}^H[x_t](s, d) &= \prod_{h_i \in (s, d)} l(h_i | x_t) \prod_{h_i \in \overline{(s, d)}} (1 - l(h_i | x_t)). \end{aligned} \quad (21)$$

All these BBA spaces are defined in the belief space. They are the belief functions distribution according to the base idea of the belief space, denoted by  $\{X_{am}(i)\}$ .

So other factors can be defined and described in the belief space, it is the base step of fusing these factors to find the optimization local target point.

**6.3.3. The Belief Function Distribution of the Optimization the Path.** In Figure 16, we give the distribution of the path belief function at one movement space. Point  $o$  is the particle of the robot, the detection distance is  $R$ , and the angle of the free space is  $\alpha$ ; thus there are two definitions of the selection of the local goal  $e$  and  $c$  for optimizing the path.

- (1) As the global target  $g_1$  is in a free space at this time,  $l_g = og_1, l_g > R, l_g = oe + eg_1, l_d = R + dg_1, l_f = of + fg_1$ , so the path proportion function in two ways is

$$f_{l_1}(l_e) = \frac{l_f - l_e}{l_f - l_g}. \quad (22)$$

The direction of the distribution  $\widehat{ef}$  is

$$\begin{aligned} f_{l_1}(l_e) &= \begin{cases} 1 & l_e = l_g \\ 0 & l_e = l_f, \end{cases} \\ f_{l_2}(l_e) &= \frac{l_d - l_e}{l_d - l_g}. \end{aligned} \quad (23)$$

The direction is  $oe$ :

$$f_{l_2}(l_e) = \begin{cases} 1 & l_e = l_g \\ 0 & l_e = l_d. \end{cases} \quad (24)$$

So the path proportion function is

$$f_l(l_e) = f_{l_1}(l_e) f_{l_2}(l_e). \quad (25)$$

- (2) As the global target  $g_1$  is out of a free space at this time,

$$l_a = oa + og_2,$$

$$l_c = \begin{cases} oc + ca + ag_2 & \text{in } a \\ oh + hg_2 & \text{out } a, \end{cases} \quad (26)$$

$$l_b = ob + bg_2,$$

$$l_h = \begin{cases} oh + ha + ag_2 & \text{in } a \\ oc + cg_2 & \text{out } a, \end{cases}$$

so the path proportion function in two ways is

$$f_{l_1}(l_c) = \frac{l_h - l_c}{l_h - l_a}. \quad (27)$$

The direction of the distribution is  $\widehat{ch}$ ; then

$$f_{l_1}(l_c) = \begin{cases} 1 & l_c = l_a \\ 0 & l_c = l_h, \end{cases} \quad (28)$$

$$f_{l_2}(l_c) = \frac{l_b - l_c}{l_b - l_a}.$$

The direction is  $ob$ , so

$$f_{l_2}(l_c) = \begin{cases} 1 & l_c = l_a \\ 0 & l_c = l_b. \end{cases} \quad (29)$$

So the path proportion function is

$$f_l(l_c) = f_{l_1}(l_c) f_{l_2}(l_c). \quad (30)$$

So the path “optimization” and “nonoptimization” plausibility function can be defined as

$$\begin{aligned} pl_{\text{dis}}^H[x_{e,c}](T, \omega) &= pl_o^H[x_{e,c}](T) f_l(l_{e,c}), \\ pl_{\text{dis}}^H[x_{e,c}](T, f) &= pl_o^H[x_{e,c}](T) (1 - f_l(l_{e,c})), \\ pl_{\text{dis}}^H[x_{e,c}](T, (\omega, f)) &= 1 - \prod_{h_i \in (\omega, f)} (1 - l(h_i | x_{e,c})). \end{aligned} \quad (31)$$

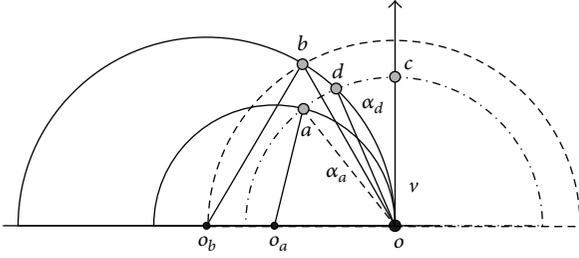


FIGURE 17: The sketch map of the belief function distribution of the dynamics of the robot at time  $t$ .

As the same way, the BBA of the “optimization” and “nonoptimization” in TBM can be defined as

$$m_{\text{dis}}^H [x_{e,c}] (\omega) = \prod_{h_i \in \omega} l(h_i | x_{e,c}) \prod_{h_i \in \bar{\omega}} (1 - l(h_i | x_{e,c})),$$

$$m_{\text{dis}}^H [x_{e,c}] (f) = \prod_{h_i \in f} l(h_i | x_{e,c}) \prod_{h_i \in \bar{f}} (1 - l(h_i | x_{e,c})),$$

$$m_{\text{dis}}^H [x_{e,c}] (\omega, f) = \prod_{h_i \in (\omega, f)} l(h_i | x_{e,c}) \prod_{h_i \in (\bar{\omega}, \bar{f})} (1 - l(h_i | x_{e,c})). \quad (32)$$

All these BBA spaces are defined in the belief space. They are the belief functions distribution according to the base idea of the belief space, denoted by  $\{X_{\text{dis}}(i)\}$ .

**6.3.4. The Belief Function Distribution of the Dynamics of the Robot.** Figure 17 shows the distribution of the dynamics of the robot at one movement space, point  $o$  is the particle of the robot, and the speed of the robot is  $v$ ; supposing the movement character is the nonglide movement, so the track of the movement is one part of the circle, the position of the local target point is  $d$ , the radius of the track is  $r_b = o_b b$ , the position  $b$  is the max distance that the robot can reach at certain time, and the min movement radius is  $r_a = o_a b$ , so the reached proportion function has two directions.

- (1) Consider the reached proportion functions in the same track radius:

$$f_{t-dg}(l_{od}) = \frac{l_{ob} - l_{od}}{l_{ob}} \quad (0 \leq l_{od} \leq l_{ob}). \quad (33)$$

- (2) Consider the reached proportion functions in the same detection area:

$$f_{t-dg}(\alpha_d) = \frac{\alpha_a - \alpha_d}{\alpha_a} \quad (0 \leq \alpha_d \leq \alpha_a). \quad (34)$$

So the reached proportion function at certain detection time at local target point  $d$  is

$$f_{dg}(l_{od}, \alpha_d) = f_{t-dg}(l_{od}) f_{t-dg}(\alpha_d). \quad (35)$$

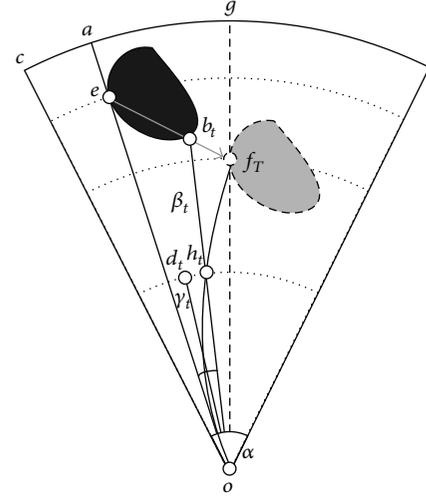


FIGURE 18: The sketch map of the belief function distribution of the escaping the movement obstacle at time  $t$ .

So the path “reach” and “unreach” plausibility function can be defined as

$$pl_{dg}^H [x_d] (T, s) = pl_o^H [x_d] (T) f_{dg}(l_{od}, \alpha_d),$$

$$pl_{dg}^H [x_d] (T, s) = pl_o^H [x_d] (T) (1 - f_{dg}(l_{od}, \alpha_d)), \quad (36)$$

$$pl_{dg}^H [x_d] (T, (s, h)) = 1 - \prod_{h_i \in (s, h)} (1 - l(h_i | x_d)).$$

Then the BBA of the “reach” and “unreach” in TBM can be defined as

$$m_{dg}^H [x_d] (s) = \prod_{h_i \in s} l(h_i | x_d) \prod_{h_i \in \bar{s}} (1 - l(h_i | x_d)),$$

$$m_{dg}^H [x_d] (h) = \prod_{h_i \in h} l(h_i | x_d) \prod_{h_i \in \bar{h}} (1 - l(h_i | x_d)), \quad (37)$$

$$m_{dg}^H [x_d] (s, h) = \prod_{h_i \in (s, h)} l(h_i | x_d) \prod_{h_i \in (\bar{s}, \bar{h})} (1 - l(h_i | x_d)).$$

All these BBA spaces are defined in the belief space. They are the belief functions distribution according to the base idea of the belief space, denoted by  $\{X_{dg}(i)\}$ .

**6.3.5. The Belief Function Distribution of the Escaping the Movement Obstacle.** Figure 18 gives the distribution of the path belief function at one movement space, point  $o$  is the particle of the robot, the detection distance is  $R$ , the angle of the free space is  $\alpha$ , the one side speed of the movement obstacle is  $v$ , and the belief function distribution of the movement obstacle in the free movement space can be defined.

The one side edge point of the obstacle  $ob$  is  $e$ , and this point can reach the position  $f_t$  after time  $T$ , so the double-safe-edges free space will be changed from  $coa$  to  $cob$ , and

the position of the robot can reach the position  $h_t$  after time  $T$ . If the position of the robot is  $h_t$  after time  $T$ , the angle is  $\beta_t$  of the  $eah_t$  and the angle is  $\gamma_t$  of the  $eah_t$ , so the it can describe the belief function distribution of the escaping the movement obstacle.

The collisions function of the robot which moves from the position  $o$  to the position  $f$  is

$$f_{\text{mov}}(\gamma_t) = \frac{\beta_t - \gamma_t}{\beta_t} \quad (0 < \gamma_t < \beta_t). \quad (38)$$

So the path “safe” and “collisions” plausibility function can be defined as

$$\begin{aligned} p_{\text{mov}}^H[x_t](T, s) &= p_o^H[x_t](T) f_{\text{mov}}(\gamma_t), \\ p_{\text{mov}}^H[x_t](T, h) &= p_o^H[x_d](T) (1 - f_{\text{mov}}(\gamma_t)), \\ p_{\text{mov}}^H[x_t](T, (s, h)) &= 1 - \prod_{h_i \in (s, h)} (1 - l(h_i | x_t)). \end{aligned} \quad (39)$$

Then, the BBA of the “safe” and “collisions” in TBM can be defined as

$$\begin{aligned} m_{\text{mov}}^H[x_t](s) &= \prod_{h_i \in s} l(h_i | x_t) \prod_{h_i \in \bar{s}} (1 - l(h_i | x_t)), \\ m_{\text{mov}}^H[x_t](h) &= \prod_{h_i \in h} l(h_i | x_t) \prod_{h_i \in \bar{h}} (1 - l(h_i | x_t)), \\ m_{\text{mov}}^H[x_t](s, h) &= \prod_{h_i \in (s, h)} l(h_i | x_t) \prod_{h_i \in \overline{(s, h)}} (1 - l(h_i | x_t)). \end{aligned} \quad (40)$$

All these BBA spaces are defined in the belief space. They are the belief functions distribution according to the base idea of the belief space, denoted by  $\{X_{\text{mov}}(i)\}$ .

**6.4. The Model of Fusing the Correlation Factors in Belief Space.** Suppose that, at any given time, the local target points set is  $Z = \{z_1, z_2, \dots, z_7\}$ , the correlation factors set is  $Q = \{q_1, q_2, \dots, q_6\}$ , and the selection state space is  $H = \{\phi, S, R, U\}$ ,  $U \neq \phi$ . For each  $z_i \in Z$ , the BBA set in the selection state space  $H$  is  $M = \{m_{q_1}^H(z_i), m_{q_2}^H(z_i), \dots, m_{q_j}^H(z_i)\}$ , so the belief space is  $X = \{X_{m_{q_1}}, X_{m_{q_2}}, X_{m_{q_3}}\}$ , and this part will combine the BBA in the belief space  $X$  to optimize the selection of the local target point.

**6.4.1. The Structure of the Local Target Point Belief Space.** Figure 19 shows three proposition spaces  $(\Omega, Z)(\Omega, Q)(\Omega, H)$  and a decision-making function; supposing that  $\Lambda_1$  is a multimapping from  $Q$  to  $Z$ ,  $\Lambda_2$  is a multimapping from  $H$  to  $Q$ ;  $\Lambda_1$  and  $\Lambda_2$  compose the “credal” level and the decision-making function composes the “pignistic” level in TBM.

In “credal” level each local target point has its own factors, so it has to filter the fusing local target point belief space to make sure of the whole factors at the same time. Each factor has its own belief space, the whole factors BBA depend on the selection state space of the factors, and this chain structure of the local target point belief space can transform the influence

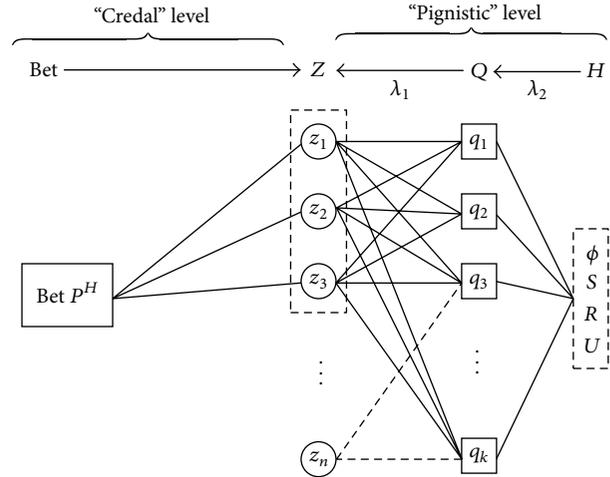


FIGURE 19: The structure of selecting the local target point belief space at time  $t$ .

of the factors to the BBA function in the belief space. In the “pignistic” level it denotes the influence degrees of the factors using the probability functions; it is the final form of selecting the local target point.

**6.4.2. The Fusing Process of the Belief Space.** It needs to fuse the belief space when the factors have been described to the BBA functions in the belief space; the details of the process at certain times are as follows.

*Step 1.* The local target points set is selected according to the double-safe-edges free space, denoted as  $Z$ , and the correlation influence factors are ascertained at certain time, denoted as  $Q$ .

*Step 2.* The BBA of the correlation influence factors set can be calculated, denoted as  $M$ , and the belief space  $X$  can be made according to the base idea of the belief space in local goal selection.

*Step 3.* In belief space  $X$ , the BBA set  $M$  can be combined according to the elements of the selection state space  $H$ , so the belief space  $X^H$  including  $M^H$  in the same state space can be made.

*Step 4.* The  $M^H$  in  $X^H$  can be transformed to the probability distribution  $\text{Bet } P^H(A)$ . For all  $A \subseteq H$ , so the optimization local target point can be selected according to the  $\text{Bet } P^H(A)$ .

## 7. The Simulation of the Local Target Point in Belief Space

In a real-time path planning process, the environment information requires to be detected at each time, so the simulation of the local target point in belief space should satisfy this character. In this paper, the maps of simulation have been made by the .bmp pictures beforehand, and the algorithm of the local target point in belief space has been written



FIGURE 20: The simulation of the path planning in special environment which has lots of edges and corners.

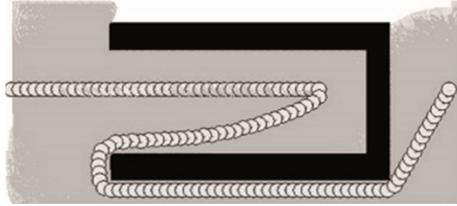


FIGURE 21: The simulation of the path planning in U-shape environment which is always called dead area.

in the software by the program, so in the process of the simulation it shows the start point and the target point, the obstacles on the map, the particle and the self-safe area, the lines of detecting, the position of the robot, and the path.

There are about two different types of simulation that will be shown in this paper to prove the feasibility of the double-safe-edges space and the idea of selecting the local target point in belief space. Simulation I is the simulation of double-safe-edges space in two conditions; it will show the special map which can show the characters of the double-safe-edges space and the death area (U shape) which can show its flexible ability. Simulation II is the simulation of efficiencies of the selecting the local target point in belief space, and it will show the changes of belief in the process of the detecting path planning.

*Simulation I.* The following two special maps show the simulation results of testing the double-safe-edges space. Figure 20 shows the special environment which has lots of edges and corners. In this environment, the sensor will be able to easily detect the edges points of the obstacles, so it is easy to transform these sensor edges points to the double-safe-edges points, and the double-safe-edges free space can be built each time, so the local target point will be found in real-time, and the robot will move to this position. From the simulation we can see that the method of double-safe-edges space has found the target point successfully and also keeps the path smoothness. Figure 21 shows the special environment which is called death area (U shape). Because of this special environment, when a robot enters this environment, there are zero obstacle edges that can be found, so it is hard to find the right local target point to escape from the obstacle. In the double-safe-edges space, this situation is the Z-DSE type, the robot can move along with one side of the obstacle until it finds the new edges of the obstacle. From the simulation we

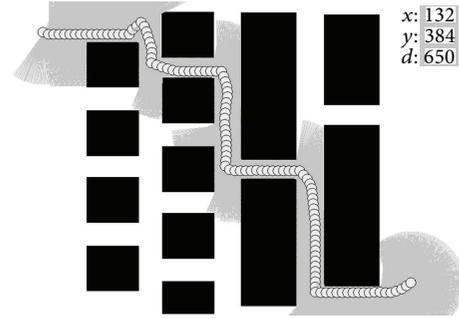


FIGURE 22: The simulation of the path in situation A for distance detecting.

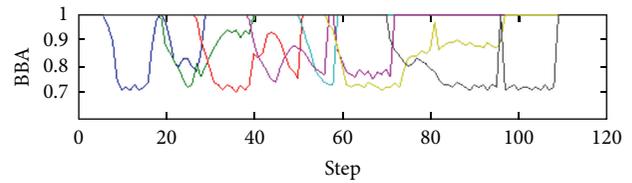


FIGURE 23: The BBA state in the process of situation A.

can see that this method can escape the death area and reach the goal successfully.

So this simulation has proved that the method of double-safe-edges space is a feasible method to describe the real-time detecting environment.

*Simulation II.* This simulation results are shown in the Figures 22 and 23. Different detection areas will make the decision belief different in belief space. Figure 22 is the situation of special A. In this saturation well the maximum detection distance is shorter than the length of the right-angle line. So in this environment the delectation ability of the sensor is low, and from the simulation we can see that the robot can reach the target point after 115 steps. Figure 23 shows the belief of 7 correlation factors of each point of these 115 points at each step, and we can see that the belief is higher the line of 0.7 belief; it means the selection of each local target point in each step shows the well belief degree.

Figure 24 is the situation of special B. In this saturation the maximum detection distance is longer than the length of the right-angle line. So in this environment the delectation ability of the sensor is high, and from the simulation we can see that the robot can reach the target point after 126 steps. Figure 25 shows the belief 7 correlation factors of each point of these 126 points at each step, and we can see that the belief is higher than the line of 0.7 belief; it means that the selection of each local target point in each step shows the well belief degree. So these simulations have proved that this belief space algorithm has the well effect.

## 8. Conclusions

As can be seen from literature works that there are a lot of methods for robot path planning, but most of them do

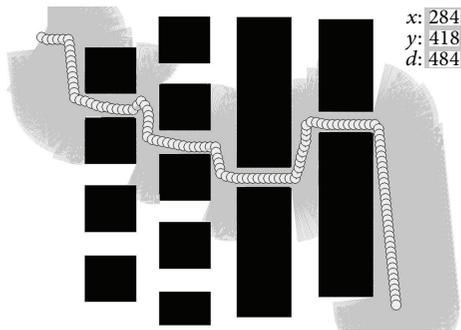


FIGURE 24: The simulation of the path in situation B for distance detecting.

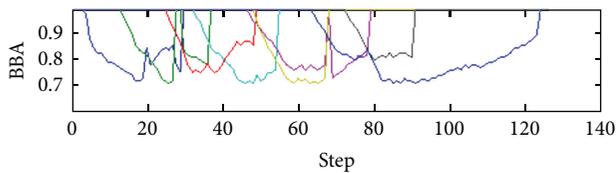


FIGURE 25: The BBA state in the process of situation B.

not work well in a complex real-time environment. In this paper, we are making some efforts for solving two problems in real-time detecting path planning: one is the expression the environment, and the second is how to optimize the path in local path planning. The double-safe-edges space has been presented to express the environment, and the simulation has proved the feasibility of this approach. Then, the belief space has fused the factors and the uncertainty of detection in real-time detecting path planning successfully, the simulation of the belief space is well running. So these achievements will help the researching of the real-time path planning effectively. Certainly, there are a lot of tough jobs such as the details of the system structure, or how to control the robot accurately. All these considerations should be further extended in our future work.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 51379049 and 51109045) and the Fundamental Research Funds for the Central Universities of China (HEUCFX41302).

## References

- [1] G. Ulivi, G. Oriolo, and M. Vendittelli, "Real-time map building and navigation for autonomous robots in unknown environments," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 28, no. 3, pp. 316–333, 1998.
- [2] L. Kleeman and K. S. Chong, "Mobile-robot map building from an advanced sonar array and accurate odometry," *International Journal of Robotics Research*, vol. 18, no. 1, pp. 20–36, 1999.
- [3] A. Buecken and S. Thrun, "Integrating grid-based and topological maps for mobile robot navigation," in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 944–950, AAAI, August 1996.
- [4] H. W. Jo, J. H. Park, G. T. Kim, and J. T. Lim, "On completeness of scalable path planning for multiple robots using graph with cycles," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 10, pp. 4167–4179, 2013.
- [5] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 116–121, 1985.
- [6] E. Ramasso, M. Rombaut, and D. Pellerin, "State filtering and change detection using TBM conflict application to human action recognition in athletics videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 944–949, 2007.
- [7] G. Ulivi, M. Poloni, and M. Vendittelli, "Fuzzy logic and autonomous vehicles: experiments in ultrasonic vision," *Fuzzy Sets and Systems*, vol. 69, no. 1, pp. 15–27, 1995.
- [8] S. X. Yang and C. M. Luo, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1279–1298, 2008.
- [9] F. C. J. Allaire, M. Tarbouchi, G. Labonté, and G. Fusina, "FPGA implementation of genetic algorithm for UAV real-time path planning," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1–3, pp. 495–510, 2009.
- [10] H. Lu, R. Y. Niu, J. Liu, and Z. Zhu, "A chaotic non-dominated sorting genetic algorithm for the multi-objective automatic test task scheduling problem," *Applied Soft Computing*, vol. 13, no. 5, pp. 2790–2802, 2013.
- [11] C.-C. Lin, "Hierarchical path planning for manipulators based on genetic algorithm with non-random initial population," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 3, pp. 1773–1786, 2012.
- [12] J. Yu and V. Kroumov, "3D path planning for mobile robots using annealing neural network," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC '09)*, pp. 130–135, March 2009.
- [13] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.
- [14] D. Mercier, É. Lefèvre, and D. Jolly, "Object association with belief functions, an application with vehicles," *Information Sciences*, vol. 181, no. 24, pp. 5485–5500, 2011.
- [15] W. Yaonan, Y. Yimin, Y. Xiaofang et al., "Autonomous mobile robot navigation system designed in dynamic environment based on transferable belief model," *Measurement*, vol. 44, no. 8, pp. 1389–1405, 2011.
- [16] B. Marhic, L. Delahoche, C. Solau, A. M. Jolly-Desodt, and V. Ricquebourg, "An evidential approach for detection of abnormal behaviour in the presence of unreliable sensors," *Information Fusion*, vol. 13, no. 2, pp. 146–160, 2012.
- [17] G. Chalkiadakis and C. Boutilier, "Sequentially optimal repeated coalition formation under uncertainty," *Autonomous Agents and Multi-Agent Systems*, vol. 24, no. 3, pp. 441–484, 2012.
- [18] R. Kuc, "Pseudoamplitude scan sonar maps," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 767–770, 2001.
- [19] G. Ulivi, G. Oriolo, and M. Vendittelli, "Fuzzy maps: a new tool for mobile robot perception and planning," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 179–197, 1997.
- [20] M. Vendittelli, F. Gambino, and G. Ulivi, "The Transferable belief model in ultrasonic map building," in *Proceedings of*

- the 6th IEEE International Conference on Fussy Systems*, vol. 3, pp. 601–608, July 1997.
- [21] A.-S. Capelle-Laizé, C. Fernandez-Maloigne, and O. Colot, “TBM for color image processing: a quantization algorithm,” in *Proceedings of the 9th International Conference on Information Fusion (FUSION '06)*, pp. 1–7, July 2006.
- [22] C. Franois and J. Allaire, “Data fusion in the transferable belief model,” in *Proceedings of the 3rd International Conference on Information Fusion*, vol. 1, pp. 21–33, 2000.
- [23] P. Smets, “Managing deceitful reports with the transferable belief model,” in *Proceedings of the 8th International Conference on Information Fusion (FUSION '05)*, vol. 2, pp. 893–899, July 2005.
- [24] P. Smets and F. Delmotte, “Target identification based on the transferable belief model interpretation of Dempster-Shafer model,” *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 34, no. 4, pp. 457–471, 2004.
- [25] P. Smets, *A model mathematical-satistical stimulating the process of diagnostic medical [Ph.D. dissertation]*, University Libre Bruxelles, Bruxelles, Belgium, 1978.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

